Improving Performance Bounds for Network min-Systems with Link Correlations

Jianwei An*, Cuiying Feng†, Kui Wu*
*Department of Computer Science, University of Victoria, Victoria, BC, Canada
†School of Information and Software Engineering, University of Electronic Science and Technology of China

***Abstract*—A network min-system is an abstract model comprising a set of min-equations, each representing the values of network links along a path. While these systems have significant practical applications, the current theoretical solutions provide estimates for variable bounds that are often too loose. This paper aims to refine these bounds using additional data derived from link correlations. Specifically, we explore two types of link correlations: fairness constraints and total capacity constraints among a node's adjacent links. We theoretically demonstrate how these correlations can enhance the performance bounds of network min-systems. Our approach systematically addresses two primary challenges in tightening these bounds: the effects of synchronous versus asynchronous updates and the cascading effect. Experimental results in the context of bandwidth tomography indicate that our algorithms advance the current state-of-the-art by significantly tightening the performance bounds of network min-systems.**

I. Introduction

A. Problem to Be Solved and Practical Implication

A network min-system refers to a group of min equations:

$$\begin{cases} x_{11} \wedge x_{12} \ldots \wedge x_{1n_1} = b_1 \\ \ldots \\ x_{m1} \wedge x_{m2} \ldots \wedge x_{mn_m} = b_m \end{cases} \quad (1)$$

where $\wedge$ denotes the min operator (i.e., $x_{11} \wedge x_{12} \ldots \wedge x_{1n} = min\{x_{11}, x_{12}, \ldots x_{1n}\}$), $b_1, \ldots, b_m$ are known values, but all x_{ij} are unknown variables governed by a network structure, e.g., $x_{11}, x_{12}, \ldots x_{1n}$ are available bandwidth of links along a path in a network. The goal is to infer the exact value x_{ij} or the upper and lower bounds of x_{ij} if the unique value cannot be determined.

Solving a network min-system has important practical meaning. In the Internet domain, monitoring the key performance indicators (KPIs), such as delay and bandwidth, is essential for making intelligent decisions regarding resource/computation scheduling [1], [2]. In a large physical network, directly measuring the performance of each link is prohibitive due to the high measurement overhead. On the Internet, a traffic flow may pass multiple Internet Service Provider(ISP) networks, and the performance of internal links is usually not visible to the public due to security concerns. In these scenarios, network tomography becomes an important technique that infers the performance or operational status of internal links of internal network based on end-to-end measurement [3]–[8]. The network min-system is the analytical model for bandwidth tomography [9].

As another example, the network min-system has also found applications in the cryptocurrency domain, e.g., in the lightning network [10]. The lightning network is an advanced technology that improves the scalability of Bitcoin and other cryptocurrencies. Transactions on the Lightning Network are facilitated through a system of channels. When two parties wish to transact, they open a payment channel between them, allowing for an unlimited number of transactions. These transactions are not individually recorded on the Bitcoin blockchain; instead, only the final state of the channel is recorded when the channel is closed. The network also supports the routing of payments across multiple channels, enabling users to send payments to each other even if they do not have a direct channel open. To support this, it is required that all balances along the payment path be higher than the transaction amount. In other words, we can formulate a min equation along the payment path. This observation has been utilized to infer the account balances in the lightning network efficiently [11].

B. Challenges and Contributions

Following the convention, when the value of a link can be uniquely determined, we call the link *identifiable* [12]. When a link is not identifiable, we turn to discover the tightest bound of each link. It is well known that solving a network min-system is extremely challenging because the min operator results in high information loss: it only tells the minimum value among the links along a path, based on which it is impossible to recover all the values on this path's constituting links. Thus, a reasonable goal is to find the tightest bound of each link determined by the min-system.

A polynomial algorithm has been designed to obtain the tightest bounds on variables in a min-system [9]. Nevertheless, the theoretically proven tightest bounds may be still too loose. Specifically, a link's upper bound is its maximum capacity. In the context of bandwidth tomography, this is equivalent to stating that the available bandwidth of a link is smaller than the link's maximum capacity, which is true but nevertheless not useful. An interesting question arises: **can we introduce extra information into the min-system to refine these bounds?**

The extra information that we consider includes two constraints: the fairness constraint and the total capacity constraint for links adjacent to a common node. The former means that the performance of these links should not be drastically

ISBN 978-3-903176-63-8

different; the latter means that the total values on these links are bounded. These constraints normally hold in network systems, with the former implying that no link should be severely over-utilized compared to other links adjacent to the common node and the latter implying that the processing capacity of a node is limited. It is worth noting that the above constraints resemble many real-world scenarios. In the Internet backbone, most routers adopt load balancing that dispatches a traffic flow among different links of the router. In this case, the bandwidth of links behind the load balancing is highly correlated. In some hybrid wired-wireless networks in the era of the Internet of Things (IoT) and edge computing [13]–[17], the performance of wireless links of a gateway node is correlated since the links share the same Radio Frequency(RF) channel.

Solving network min-system with correlated links poses several non-trivial difficulties: First, one basic assumption in existing work that links are independent does not hold anymore. Second, assuming bi-directional links[1] may not be appropriate. For instance, in the wireless scenario, the uplinks and downlinks may use different RF channels and thus form different interference link sets. Third, an update on one link's performance may generate a cascading effect on other links' performance, and we might obtain different final results when updating link performance in different orders.

This paper pushes the horizon of network min-system research by addressing all the above challenges. To be more specific, we make the following contributions:

1) We present a new network model that consists of bi-directional links and formally formulate constraints that capture the correlation among local links. We investigate how the correlation constraints help narrow down the bounds.
2) We develop a polynomial time algorithm, called Global ϵ-stabilizing (GES) algorithm, to calculate the best-effort bounds in network min-systems with link correlation constraints. We also work out the sufficient condition where the best-effort bounds are the tightest bound.
3) We extensively evaluate GES and compare its performance with that of the Calculate the Tightest Bounds(CTB) benchmark [9]. Evaluation results show that GES achieves much tighter performance bounds than CTB.

II. Related Work

The concept of network tomography was first introduced in [3]. Since then, network tomography has been extensively studied, with the goal of inferring internal network performance based on end-to-end measurements. Most work on network tomography focuses on additive metrics [4]–[7], [18]–[25]. Network tomography involving a min-system was recently studied in [9]. A min-system in the cryptocurrency domain for account balance inference can be found in [11].

[1]A bi-directional link between node A and node B means that the performance from node A to node B is equal to that from B to A. Otherwise, we call the link uni-directional.

Extensive research has been devoted to end-to-end bandwidth estimation [26], and many well-developed measurement tools have been developed for this purpose, e.g., *pathchar*, *clink*, *pchar*, and *bfind*. These works, however, can only formulate each min equation rather than solve the min-system.

The authors of [9] provided a polynomial algorithm called CTB to obtain the optimal solution of a min-system. Nevertheless, the optimal solution is based on the assumption that no link correlation information is available. This paper pushes the state-of-the-art by leveraging link correlation to further improve the performance bounds for a min-system.

III. System Model

We present a generic abstract network model that captures the link correlation that exists in many scenarios. The network is modelled as a directed graph $\mathcal{G} = (V, L)$, where V denotes the set of nodes (e.g., computers or routers), L denotes the set of links between nodes. Note that for two nodes v_1 and v_2, the performance value from v_1 to v_2, denoted by l_{12}, may be different from that from v_2 to v_1, denoted by l_{21}. We call the link between v_1 and v_2 bi-directional when $l_{12} = l_{21}$, otherwise, we call it uni-directional. In summary, $\mathcal{G}$ is a connected and directed graph. Each link has distinct end nodes (i.e., no self-loop), and no two links of the same direction in $\mathcal{G}$ connect to the same pair of nodes. We explain our system model using bandwidth tomography as the application context to ease discussion.

With a set of monitors deployed at some nodes in the network, we assume that we can measure the end-to-end performance of a *Measurement path (MP)*. Here, an MP refers to a non-loop path that only contains two monitors at its end nodes. We can use existing methods, such as *pathload* [27] and RT-WABest [28], to measure the end-to-end available bandwidth of an MP. Here, we are interested in inferring the available bandwidth (or **bandwidth for short**) of all links in the network based on measurement results along MPs. To simplify the analysis, we assume that the maximum bandwidth over all links is b_{max}. Note that the analytical results of this paper are easily applicable to the scenario where different links have different maximum bandwidth values. The only change is that each link uses its own maximum value. Initially, b_{max} could be set to the physical limit of the link based on the hardware specification.

The bandwidth values in a local environment may be correlated. As shown in Fig. 1, if routers v_5 and v_6 are equipped with a load balancer to balance the traffic load towards the connected servers, then l_{51} and l_{52} are correlated, and l_{63} and l_{64} are also correlated. Suppose that our network under analysis is a hybrid wireless network that consists of both wired and wireless links. In a local area (e.g., base station), the wireless uplinks may share the same radio channel and thus their bandwidth values are correlated; similarly, the wireless downlinks may share another radio channel and thus form another set of correlated links [29]. We next propose constraints that can be used to analyze all the above scenarios

where links sharing common nodes are correlated. For this purpose, we consider two constraints that have broad practical implications. To simplify notation, we use the same notation to denote a link and the bandwidth value of the link.

Definition 1. ***Total capacity constraint:*** *For a subset of links connected to a node v, called correlation set and denoted by $\{l_1^v, l_2^v, \ldots, l_m^v\}$, we assume that:*

$$\sum_{i=1}^{m} l_i^v \leq b_{max}^v. \tag{2}$$

This constraint is based on the observation that a router's total packet processing speed is limited. This constraint is also applicable to the wireless case, where the wireless links share the same radio channel, and their total bandwidth should be no larger than the capacity of the channel.

Definition 2. ***Fairness constraint:*** *For a subset of links connected to a node v, called correlation set and denoted by $\{l_1^v, l_2^v, \ldots, l_m^v\}$, we assume that:*

$$\mathcal{J}(l_1^v, \ldots, l_m^v) \equiv \frac{(\sum_{i=1}^m l_i^v)^2}{m \sum_{i=1}^m (l_i^v)^2} \geq \delta_v. \tag{3}$$

Eq. (3) means Jain's fairness index [30] is higher than a threshold, reflecting the fact that load balancing in a router tries to avoid overloading a particular outgoing link. This constraint is also reasonable to wireless networks since most wireless systems have mechanisms to guarantee a certain level of fair share of the channel among the local wireless links [31], [32]. Note that Jain's fairness index is a value between 0 and 1, and a higher value means more fair. If the index is 1, all links (in the correlation set) have the same value. As the link disparity increases, the index decreases.

Remark 1. *The constraints are optional, i.e., they are posed only to the nodes that we know have these constraints. In addition, for a given node v, the correlation set for the total capacity constraint and the correlation set for the fairness constraint are not necessarily the same.*

Remark 2. *We assume that the network under consideration is "static", implying that either the bandwidth changes slowly relative to the measurement process or it represents statistical characteristics (e.g., mean) that stay constant within the time period under consideration. This assumption has been broadly adopted in most network tomography work [5], [7], [23], [33]. It has been observed that the available bandwidth of wireless links can be considered stationary [34] over a short time period, even if, in general, it is dynamic Quality of Service(QoS) information.*

In the following, we use a simple network in Fig. 1 to illustrate the network min-system with correlation constraints. Assume that three monitors are deployed at nodes v_1, v_2 and v_3, respectively. There are six possible MPs in total, which are (1) $v_1 \to v_5 \to v_2$, (2) $v_2 \to v_5 \to v_1$, (3) $v_1 \to v_5 \to v_6 \to v_3$, (4) $v_3 \to v_6 \to v_5 \to v_1$, (5) $v_2 \to v_5 \to v_6 \to v_3$, and (6) $v_3 \to v_6 \to v_5 \to v_2$. Assume that the end-to-end available bandwidth of the above 6 MPs is $b_1, b_2, \ldots, b_6$, respectively.

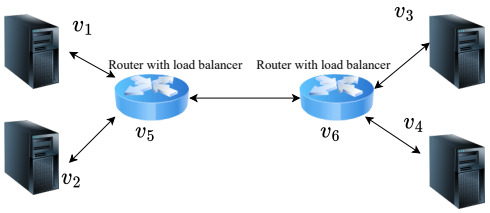


Fig. 1: An example network with correlated links caused by load balancing. v_1, v_2, and v_3 are also monitoring nodes.

Since the bandwidth of a path is the minimum bandwidth of all links along the path, we have the following system of min-equations:

$$\begin{cases} l_{15} \wedge l_{52} = b_1 \\ l_{25} \wedge l_{51} = b_2 \\ l_{15} \wedge l_{56} \wedge l_{63} = b_3 \\ l_{36} \wedge l_{65} \wedge l_{51} = b_4 \\ l_{25} \wedge l_{56} \wedge l_{63} = b_5 \\ l_{36} \wedge l_{65} \wedge l_{52} = b_6 \end{cases} \tag{4}$$

where l_{ij} denotes the bandwidth from node v_i to node v_j and $\wedge$ means the min operation.

With a slight abuse of notation $\wedge$, let's denote the above linear system into *equivalent* matrix form $\mathbf{R} \wedge \mathbf{L} = \mathbf{B}$, where

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{5}$$

$$\mathbf{L} = \left(l_{15}, l_{51}, l_{25}, l_{52}, l_{36}, l_{63}, l_{46}, l_{64}, l_{56}, l_{65}\right)^\intercal, \tag{6}$$

$$\mathbf{B} = \begin{pmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{pmatrix}^\intercal. \tag{7}$$

If we know v_5 poses load balancing on links l_{52} and l_{51} and v_6 poses load balancing on links l_{63} and l_{64}. We then have the following fairness constraints due to the load balancing in v_5 and v_6:

$$\begin{cases} \mathcal{J}(l_{51}, l_{52}) \geq \delta_{v_5} \\ \mathcal{J}(l_{63}, l_{64}) \geq \delta_{v_6} \end{cases}$$

If we know that v_5 has a control mechanism that limits the total bandwidth to/from v_1 and v_2, and v_6 has a control mechanism that limits the total bandwidth to/from v_3 and v_4, we have the following total capacity constraints:

$$\begin{cases} l_{51} + l_{52} \leq b_{max}^{v_5\downarrow} \\ l_{63} + l_{64} \leq b_{max}^{v_6\downarrow} \\ l_{15} + l_{25} \leq b_{max}^{v_5\uparrow} \\ l_{36} + l_{46} \leq b_{max}^{v_6\uparrow} \end{cases}$$

where different links' total capacities differentiate $b_{max}^{v_i\downarrow}$ from $b_{max}^{v_i\uparrow}$. For a link l, if we can determine its exact value, we call the link identifiable. Otherwise, we can only determine an interval, also called error bound, that covers its value. This paper aims to answer the following fundamental problem: **given $\mathbf{R}$, $\mathbf{B}$ and the aforementioned constraints, can we infer the exact values or the tightest error bounds of $\mathbf{L}$?**

TABLE I: Main Notations

Notation	Explanation
$\mathcal{L}_{all}$	The set of all the links in the min-system
$\mathcal{N}_{all}$	The set of all the nodes that are passed by at least one measurement path
$\mathcal{L}_v$	The set of all links involved in the same constraints of specific (8) or (9) at node v
$\mathcal{L}_v^I$	The set of all the current identifiable links in $\mathcal{L}_v$ at node v
$\mathcal{L}_v \setminus \mathcal{L}_v^I$	The set of all the current unidentifiable links in $\mathcal{L}_v$ at node v
α_v	A parameter defined in (12) for convexity check at node v

* Other notations are omitted to save space.

IV. Bound Analysis and Algorithm for Obtaining the Tightest Error Bound

A. Elaboration on the Tightest Error Bound

In [9], an algorithm called CTB was proposed to calculate the "tightest" error bounds of $\mathbf{L}$. Nevertheless, since the CTB algorithm assumes no link correlation information is available and does not change the upper bound of link values, CTB returns the highest lower bound based only on the set of min-equations. Nevertheless, the new link correlation information offers us new opportunities to further reduce the error bound, i.e., raising the lower bound and decreasing the upper bound of an unidentifiable link. In addition, some links may become identifiable due to these additional constraints. Accordingly, we should first make clear the exact meaning of the tightest error bound in our new context.

Definition 3. ***(Tightest error bound)****: In the network min-system with correlated links, the tightest error bound of an unidentifiable link is defined as the smallest interval containing all possible values the link may have, satisfying the min-system and all the correlation constraints.*

B. Challenges in Obtaining the Tightest Error Bounds

Obviously, considering all links' correlation constraints at the same time complicates the solution. It's worth noting that each correlation constraint only applies to the links associating a local node. Thus, firstly, we should investigate how the bound information of an unidentifiable link l_1^v could be influenced by the correlation constraints associated with node v.

Clearly, to obtain the tightest error bound for l_1^v, we need to seek its worst-case minimum value and the worst-case maximum value using the directly-related correlation constraints. In other words, if putting the min-equations aside, we need to solve the following two optimization problems (8) and (9), respectively.

$$\begin{aligned} \min \quad & l_1^v \\ \text{s.t.} \quad & \frac{(\sum_{i=1}^m l_i^v)^2}{m\sum_{i=1}^m {l_i^v}^2} \geq \delta_v, \\ & \sum_{i=1}^m l_i^v \leq b_{max}^v, \\ & l_i^v \in [\hat{l}_{i-}^v, \hat{l}_{i+}^v], i = 1, \cdots, m, \end{aligned} \tag{8}$$

and

$$\begin{aligned} \min \quad & -l_1^v \\ \text{s.t.} \quad & \textit{the same constraints as } (8), \end{aligned} \tag{9}$$

where $l_i^v (i = 1, \cdots, m)$ are all links having constraints associated with node v, the fairness parameter is δ, the total capacity parameter is b_{max}^v, and $\hat{l}_{i-}^v$ and $\hat{l}_{i+}^v$ are link l_i^v's *current* lower bound and upper bound, respectively. Then, we take the min-system into consideration. Initially, the *current* lower and upper bounds can be obtained with the CTB algorithm [9].

Essentially, the solutions to Problems (8) and (9) will be the basic building blocks to infer the tightest error bounds for all unidentifiable links. Even having the above building blocks, however, we still need to answer two challenging problems:

1) **Challenge 1:** Synchronous vs asynchronous updates. For a given link, we can update its lower and upper bounds synchronously (i.e., solve Problems (8) and (9) in parallel) or asynchronously (i.e., solving Problems (8) and (9) in sequential. When a node is connected with multiple links, we can update the upper and lower bounds of these links synchronously (i.e., update all links' bounds in parallel) or asynchronously (update them sequentially). Will different ways of updating bounds lead to different answers for links associated with the same node?
2) **Challenge 2:** Cascading impact: In the whole network, when a node updates the bounds of its associated links, the results may impact the link-bound updates of other nodes. Will the update order among different nodes lead to different final answers?

It is worth noting that Challenge 1 concerns the bound update order *locally*, i.e., updating links associated with a local node, and Challenge 2 concerns the bound update order *globally*, i.e., the order for node-by-node update. In the following, we theoretically analyze the condition that leads to unique, tightest bounds for local updates resilient to synchronous/asynchronous updates. We design an algorithm that leads to the guaranteed tightest error bounds when the condition holds and can empirically improve the error bounds of CTB (Cf. Section V) when the condition does not hold.

C. Tackling Challenge 1

Since the total capacity constraint is linear, the main difficulty of solving Problems (8) and (9) comes from the fairness

Algorithm 1: CBLN(l_i, v): Calculating bound for an unidentifiable Link l_i at Node v

input : δ_v, b^v_{max}, $\mathcal{L}_v(|\mathcal{L}_v| = m)$, $\mathcal{L}^I_v$ ($|\mathcal{L}^I_v| = n$). For each $l_k \in \mathcal{L}_v \setminus \mathcal{L}^I_v$, its current bound $[\hat{l}_{k-}, \hat{l}_{k+}]$.

output: $\mathcal{C}_{l_{i-}}(v) = \{l^-_{v(j,i)} | l_j \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}^I_v\}$ where $l^-_{v(j,i)}$ is the value that l_j took when $l_i = \hat{l}'_{i-}$ by its (8) at v; $\mathcal{C}_{l_{i+}}(v) = \{l^+_{v(j,i)} | l_j \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}^I_v\}$ where $l^+_{v(j,i)}$ is the value that l_j took when $l_i = \hat{l}'_{i+}$ at v;The new obtained bandwidth bound $[\hat{l}'_{i-}, \hat{l}'_{i+}]$

begin
  $\alpha_v = \delta_v m - m + n$;
  if $\alpha_v \geq 0$ **then**
    Obtaining the eligible numerical approximated global optimal solution $\hat{l}'_{i-}$ of (8) of l_i at v and formulating corresponding $\mathcal{C}_{l_{i-}}$;
    Obtaining the eligible numerical approximated global optimal solution $\hat{l}'_{i+}$ of (9) of l_i at v, formulating corresponding $\mathcal{C}_{l_{i+}}$;
  else
    $\hat{l}'_{i-} = \hat{l}_{i-}$;
    $\hat{l}'_{i+} = \hat{l}_{i+}$;
    $l_{new} = b^v_{max} - \sum_{l_k \in \mathcal{L}^I_v} I_k - \sum_{l_k \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}^I_v} \hat{l}_{k-}$;
    if $l_{new} < \hat{l}_{i+}$ **then**
      $\hat{l}'_{i+} = l_{new}$;
      $\mathcal{C}_{l_{i+}}(v) = \{l^+_{v(k,i)} = \hat{l}_{k-} | l_k \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}^I_v\}$;
    end
  end
end

constraint. We can transform the fairness constraint into the following form:

$$f(l^v_1, ..., l^v_m) = \delta_v m({l^v_1}^2 + ... + {l^v_m}^2) - (l^v_1 + ... + l^v_m)^2 = \mathbf{l}^T H \mathbf{l} \leq 0, \quad (10)$$

where $\mathbf{l} = (l^v_1, ..., l^v_m)^T$, $m \geq 2$, and

$$H = \begin{pmatrix} \delta_v m - 1 & -1 & -1 & \dots & -1 \\ -1 & \delta_v m - 1 & -1 & \dots & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & -1 & \dots & \delta_v m - 1 \end{pmatrix}.$$

H is an $\mathbf{R}^{m \times m}$ symmetric matrix. The elements of H are all -1 except that the diagonal elements are all $\delta_v m - 1$. We ignore the case $m = 1$ because it is meaningless to discuss fairness if there is only one link.

Note that during the process of bounds update, an unidentifiable link may become identifiable. To ease discussion with a unified presentation, we extend the constraint (10) into a more general form that reflects the fairness constraint changes during the bounds update process. To be specific, suppose $\mathcal{L}_v$ consists of all the links involved in the same constraints of (8) or (9) at node v and $|\mathcal{L}_v| = m$, when $n(1 \leq n \leq m-1)$ links become identifiable at node v, we record these links in set $\mathcal{L}^I_v$ and denote their corresponding values as $I^v_k (k = 1, \cdots, n)$. We also record the rest unidentifiable links in set $\mathcal{L}_v \setminus \mathcal{L}^I_v$. The fairness constraint (10) the links in $\mathcal{L}_v \setminus \mathcal{L}^I_v$ could be revised to

$$f(l^v_1, ..., l^v_{m-n}) = \mathbf{l}^T H' \mathbf{l} - 2 S_1 \mathbf{l}^T \mathbf{1} + \delta_v m S_2 - S_1^2 \leq 0, \quad (11)$$

where $\mathbf{l}^T = (l^v_1, ..., l^v_{m-n})$, $S_1 = \sum_{l^v_i \in \mathcal{L}^I_v} I^v_i$, $S_2 = \sum_{l^v_j \in \mathcal{L}^I_v} {I^v_j}^2$, and H' is an $\mathbf{R}^{(m-n)\times(m-n)}$ symmetric matrix whose elements are all -1 except the diagonal all $\delta_v m - 1$.

Because $|H' - \lambda I| = (\delta_v m - \lambda)^{m-n-1}(\delta_v m - m + n - \lambda)$ (we omit its derivation for brevity), the eigenvalues of H' are $\delta_v m$ and $\delta_v m - m + n$. Denote:

$$\alpha_v = \delta_v m - m + n. \quad (12)$$

When $\alpha_v \geq 0$, H' is a positive semi-definite matrix, and Problems (8) and (9) for all constrained links are all **convex**. In this case, existing convex programming tools (i.e. GUROBI, MOSEK) could help search the globally unique optimal solutions under given accuracy in polynomial time.

Based on the above analysis, we design Algorithm 1(CBLN(l_i, v)), which calculates the new bound for a link l_i at node v. Essentially, when $\alpha_v \geq 0$, CBLN(l_i, v) updates the bounds using both the fairness and total capacity constraints; otherwise, it updates the bounds using the total capacity constraint. Then, we have the following lemma:

Lemma 1. *Assume that the bounds information of all links in $\mathcal{L}_{all}$ are fixed. $\forall l_i, l_j \in \mathcal{L}_v \setminus \mathcal{L}^I_v$, the output of CBLN($l_i$, v) would not impact the output of CBLN(l_j, v).*

Lemma 1 means that the algorithm 1 is resilient to the order of link updates associated with a local node. While we omit its formal proof due to the space limit, it is not hard to understand why Lemma 1 holds: when $\alpha_v \geq 0$, we utilize the convexity to obtain the unique optimal solution; otherwise, we only use the total capacity constraint (i.e., link constraint). Due to the good property of the algorithm 1 at a specific node v, we successfully tackled Challenge 1.

D. Tackling Challenge 2

The CBLN algorithm primarily targets the bound calculation for a single link connected to a node. Note that the definition of the tightest error bound considers all the nodes' correlation constraints. The output of CBLN at node v may influence some CBLN processes of a neighbouring node v'. To explain why, assume that nodes v and v' are connected with a link l. When we update the bound of link l using the constraints associated with node v, this update may substantially influence the updates of other links due to the constraints associated with node v'. This "ping-pong" effect for bound updates needs to be carefully managed to prevent a perpetual cycle of updates over the whole min-system, ultimately hindering our ability to reach a conclusion when we want to obtain the tightest bound for all the involved links.

To avoid the above problem and reduce the redundant calculation at the same node v, we introduce a threshold ϵ and design the ϵ-LUAN(v) algorithm for locally updating all links associated with a node. The main idea is that when the bound improvement of one link is smaller than ϵ, its further changes will not be propagated to other links' bound updates. The details of the ϵ-LUAN algorithm can be found in Appendix.

So far, we have worked out how to update a group of links at a specific node v. We call this step the *node update* step, meaning that all links associated with this node are updated. Next, we further study the inter-impact between node updates, i.e., the cascading impact. To ease discussion, we call a **link** l **stable** at node v if running ϵ-LUAN(v) would not change l's bound. We call a **node** v **stable** if all links associated with v are stable.

It is worth noting that a stable node v may become unstable after when updating its neighbouring node u. This is the main cause of the "ping-pong" effect over the whole min-system with correlation constraints. Therefore, we should make as many stable nodes as possible. For this, it would be important to keep track of the returning-unstable nodes after running each ϵ-LUAN. The following procedure helps us to keep track of all unstable nodes due to running ϵ-LUAN(v).

Procedure 1. *In the given min-system with correlation constraints, suppose bounds information of all links in $\mathcal{L}_{all}$ are currently fixed, $\mathcal{N}_v^A$ consists of all the nodes that are connected to v by a current unidentifiable bidirectional link and $v \in \mathcal{N}_{all}$ is a given node.*

Suppose that $v_i \in \mathcal{N}_v^A \cap \mathcal{N}_{all}$ is connected to v by l_a, and we ran ϵ-LUAN(v_i) before. $\forall l_b \in \mathcal{L}_{v_i} \setminus \mathcal{L}_{v_i}^I (l_b \neq l_a)$, $l^-_{v_i(a,b)}, l^+_{v_i(a,b)}$ are stored according to the latest ϵ-LUAN(v_i). And before this tth running ϵ-LUAN(v), $[\hat{l}_{a-}, \hat{l}_{a+}]$ is the current bound of l_a. During this tth ϵ-LUAN(v), we obtain the returned $[\hat{l}'_{a-}, \hat{l}'_{a+}]$ from CBLN(l_a, v).

When $|\hat{l}'_{a-} - \hat{l}_{a-}| + |\hat{l}'_{a+} - \hat{l}_{a+}| \geq \epsilon$, v_i would be collected into $\mathcal{N}^N_{v(t)}$ if it would satisfy at least one of following conditions:

- *$\exists l_b \in \mathcal{L}_{v_i} \setminus \mathcal{L}^I_{v_i}$, $\hat{l}'_{a-} > l^-_{v_i(a,b)}$ or $\hat{l}'_{a+} < l^+_{v_i(a,b)}$;*
- *$\exists l_b \in \mathcal{L}_{v_i} \setminus \mathcal{L}^I_{v_i}$, $l_b \in \mathcal{L}^I_{v(t)}$.*

$\mathcal{L}^I_{v(t)}$ consists of all the identifiable links caused by this tth ϵ-LUAN(v) and $\mathcal{N}^I_{v(t)}$ collects all the nodes which control at least one link in $\mathcal{L}^I_{v(t)}$. All the nodes that possibly become unstable due to running tth ϵ-LUAN(v) on l_a are collected in $\mathcal{N}^{unstable}_{v(t)} = \mathcal{N}^I_{v(t)} \cup \mathcal{N}^N_{v(t)}$.

Utilizing the above procedure, we develop our GES algorithm(Algorithm 2) to ensure the eventual stabilization of all nodes in $\mathcal{N}_{all}$. To explain the termination of the GES algorithm, we highlight a crucial observation: when a stable node v reverts to an unstable state, it indicates the potential for further reduction in the bounds of some links connected to this node. This implies that the error bound of a link consistently diminishes over time. Importantly, this reduction process is inherently finite with given ϵ, as the error bound cannot be decreased indefinitely. Consequently, the GES algorithm eventually terminates, guaranteeing the stability of all nodes in the system.

Remark 3. *The GES algorithm is built over the ϵ-LUAN algorithm, which calls the CBLN algorithm. GES is the final algorithm that returns the performance bounds of the network min-system with correlation constraints.*

Complexity Analysis of GES: Given a min-system with correlation constraints, in the worst case, $t_\epsilon = 2|\mathcal{L}_{all}| + |\mathcal{N}_{all}| + \sum_{l_i \in \mathcal{L}_{all}} \lceil \frac{\hat{l}_{i+} - \hat{l}_{i-}}{\epsilon} \rceil$ is the maximum number of nodes which require ϵ-LUAN algorithm during the whole GES process. The time complexity of CBLN and ϵ-LUAN is in the same order as solving the convex optimization Problems (8) and (9) when $\alpha_v \geq 0$ or linear when $\alpha_v < 0$.

Algorithm 2: GES: Global ϵ-stabilizing algorithm on $\mathcal{N}_{all}$

```
input : N_all
output: The performance bounds of all links in L_all.
begin
    Initialize updating queue Q_update and add all nodes
      in N_all to Q_update;
    while Q_update is not empty do
        v_i ← Q_update.pop(0);
        Running ε-LUAN(v_i);
        According to Procedure 1, obtain corresponding
          N^unstable_{v_i(t)};
        Add all nodes in N^unstable_{v_i(t)} to Q_update.
    end
end
```

Finally, we summarize the property of the GES algorithm with lemma 2, whose proof is omitted to save space.

Lemma 2. *Given a min-system with correlation constraints, assume that $\forall l_i \in \mathcal{L}_{all}, [l_{i-}, l_{i+}]$ denotes its tightest error bound (i.e., the ground-truth). Suppose that $[l^G_{i-}, l^G_{i+}]$ is the GES-returned bound for l_i with certain ϵ,*

- *Correctness: $[l_{i-}, l_{i+}] \subseteq [l^G_{i-}, l^G_{i+}]$;*
- *Conditional Optimality: If $\forall v \in \mathcal{N}_{all}$ equipped with fairness constraint has $\alpha_v \geq 0$, then when $\epsilon \to 0$, $l^G_{i-} \to l_{i-}$ and $l^G_{i+} \to l_{i+}$.*

V. Performance Evaluation

A. Experimental Setup

We compare our GES algorithm with the state-of-the-art CTB algorithm [9] for solving a min-system. We test them over real-world ISP networks. In each network, we randomly select monitoring nodes and adopt the GSPC algorithm [9] to construct measurement paths among those monitoring nodes. In each scenario, we simulate the behaviour of load balancers at randomly selected routers by setting the available bandwidth values among the links associated with a selected router to satisfy fairness constraints. Note that we require that a selected router should have at least two associated links because otherwise, the load balancer takes no effect. In the rest, we

use $\mathcal{N}_{LB}$ to record the routers that pose fairness constraints. Regarding total capacity constraints, we sum up the maximum bandwidth values of all links associated with a router and set the total value as the constraint. Note that in practice, a link's maximum bandwidth value is available based on hardware specification.

We tested over four real-world topologies, including Abovenet (AS6461), EBONE (AS1755), Exodus (AS3967), and Tiscali (AS3257), whose topologies are from the Internet Topology Zoo (http://www.topology-zoo.org). The network parameters are listed in Table II.

TABLE II: Network Parameters

ISP name	$\|L\|$	$\|V\|$	$\|\mathcal{N}_{LB}\|$	R_{δ_v}	R_b	$\|\mathcal{M}\|$
Abovenet	294	182	82	$[0.90, 0.95]$	$[20, 300]$	82
EBONE	381	172	72	$[0.90, 0.95]$	$[20, 300]$	37
Exodus	434	201	100	$[0.90, 0.95]$	$[20, 300]$	39
Tiscali	404	240	120	$[0.90, 0.95]$	$[20, 300]$	90

δ_v denotes the threshold of fairness constraint of node v, R_{δ_v} the possible range where fairness threshold δ_v would be sampled, R_b the range of bandwidth values from which the ground-truth values are sampled, and $|\mathcal{M}|$ the number of monitors.

We use the total error bound (TEB), which is defined as the sum of error bounds of all links in the min-system, to compare the performance of different methods.

B. Performance Results

In all four real-world network topologies, finding out all the possible measurement paths between monitors is impractical (the problem of listing all MPs is NP-hard). With the GSPC algorithm, we would build effective MPs that help reduce the TEB. The number of measurement paths built with GSPC $|\mathcal{P}_{GSPC}|$ for different networks are displayed in Table III.

For each network scenario with a given ϵ, we run GES 100 times, each with a random node update sequence, and record the smallest TEB as the final result. Note that the node update sequence and ϵ have no impact on CTB, so we only need to run CTB once for each scenario. The code is implemented in Python 3.11, and the experiments are executed on a laptop (6-Core Intel Core i7, 2.2 GHz, MEM: 16 GB, macOS Sonoma Version 14.4.1). Table III shows the performance comparison results between GES and CTB when ϵ is set to 0.1. The results demonstrate that with link correlation constraints, GES can significantly reduce the TEB, leading to over 70% TEB reduction over the CTB algorithm. In addition, GES can obtain more identifiable links than CTB.

TABLE III: Performance Comparison between GES and CTB

ISP name	$\|\mathcal{P}_{GSPC}\|$	$\|\mathcal{I}\|$ (GES/CTB)	TEB (GES/CTB)	TEB reduction(%)
Abovenet	5000	**123**/108	**4789.0**/16894	72%
EBONE	5000	**140**/127	**9079.4**/32139	72%
Exodus	6000	**124**/114	**12604.4**/49063	74%
Tiscali	6000	**155**/125	**11482.2**/47227	76%

$|\mathcal{P}_{GSPC}|$ denotes the number of measurement paths built with GSPC. $|\mathcal{I}|$(GES/CTB) denote the number of identifiable links obtained with GES and CTB, respectively.

C. The Impact of ϵ

As we have mentioned before, there is interdependence among the links updates, and a minor adjustment to a link's bound may substantially influence the updates of other links in the min-system. We have used parameter ϵ to control this "ping-pong" effect. As shown in Figure 2, the fluctuation displayed from different node update sequences reveals that the potential links' improvements discarded by ϵ-LUAN at different nodes lead to different final results due to the "ping-pong" effects over the whole min-system. Nevertheless, with the decrease of ϵ, not only does the TEB reduce, but also the TEB fluctuates in a smaller range.

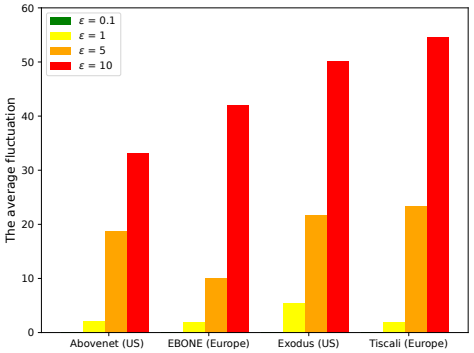


Fig. 3: The average fluctuation under different ϵ values over 100 GES runs.

To further evaluate the impact of ϵ on the final results. We run GES multiple times, each run using a random node update sequence. We define a metric, average fluctuation, to account for the distance between the TEB of different runs and the minimum TEB obtained so far. That is, given ϵ and N runs of the GES algorithm, the average fluctuation is calculated as:

$$\frac{\sum_{i=1}^{N}(TEB_{(\epsilon,i)} - TEB_{est})}{N} \tag{13}$$

where $TEB_{(\epsilon,i)}$ is the TEB given ϵ at the ith run of GES and $TEB_{est} = \min_i TEB_{(\epsilon,i)}$. Figure 3 shows that the average fluctuation decreases as the decreasing of the ϵ value. When $\epsilon = 0.1$, the final results remain nearly unchanged.

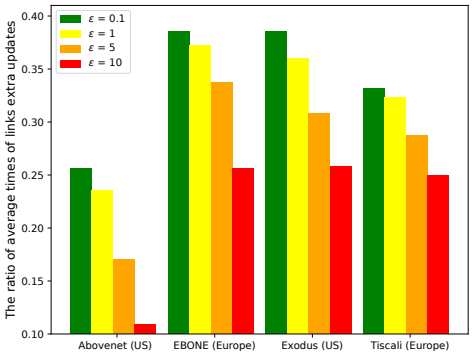
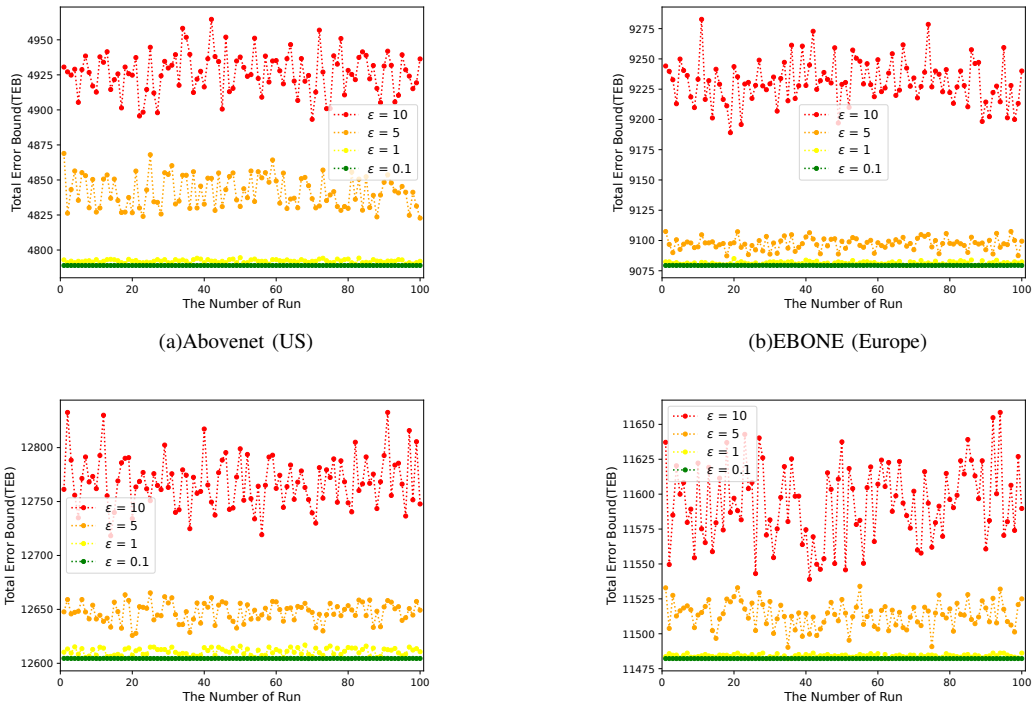


Fig. 4: The ratio of the average times of links' extra updates under different ϵ values over 100 GES runs.



(a)Abovenet (US)

(b)EBONE (Europe)

(c) Exodus (US)

(d) Tiscali (Europe)

Fig. 2: The performance of GES over different update sequences with different ϵ values.

Figure 4 shows the ratio of the average times of links' extra updates, which is calculated as:

$$\frac{\sum_{i=1}^{N}(T_{(\epsilon,i)} - |\mathcal{L}_{unident}^{CTB}|)}{N|\mathcal{L}_{unident}^{CTB}|} \tag{14}$$

where $T_{(\epsilon,i)}$ is the total times of links updated given ϵ at the ith run of GES and $\mathcal{L}_{unident}^{CTB}$ is the set of unidentifiable links only derived from the CTB results of the corresponding min-system. Clearly, a smaller updating threshold ϵ leads to more frequent extra link updates. Figures 3 and 4 together indicate the tradeoff between the quality of performance results and the computation overhead.

VI. Conclusion

This paper pushed the state-of-the-art performance bounds of network min-systems by leveraging link correlations, specifically through fairness and total capacity constraints. It improved upon the theoretically optimal bounds established by CTB [9], which did not account for link correlation information. Our work demonstrated that incorporating link correlation information, despite the intuitive appeal, presents significant challenges. Notably, the choice between synchronous and asynchronous updates over the whole min-system can yield divergent bounds, while the intricate nature of link correlations can induce a cascading effect in bound adjustments, a phenomenon we described as the "ping-pong" effect.

Addressing these complexities, we developed a Global ϵ-Stabilizing (GES) algorithm. It is designed to be resilient against synchronous/asynchronous updates and ensures that the results' fluctuations can be strictly controlled using a predefined threshold, ϵ. Through extensive simulations on real-world ISP topologies, the GES algorithm not only significantly tightens the performance bounds of CTB but also returns more identifiable links, i.e., the links whose value is uniquely determined.

References

[1] Y. Breitbart, C.-Y. Chan, M. Garofalakis, R. Rastogi, and A. Silberschatz, "Efficiently monitoring bandwidth and latency in ip networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 2. IEEE, 2001, pp. 933–942.

[2] K. Suh, Y. Guo, J. Kurose, and D. Towsley, "Locating network monitors: complexity, heuristics, and coverage," *Computer Communications*, vol. 29, no. 10, pp. 1564–1577, 2006.

[3] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

[4] E. Lawrence, G. Michailidis, V. Nair, and B. Xi, "Network tomography: a review and recent developments," *Ann Arbor*, vol. 1001, no. 48, pp. 109–1107, 2006.

[5] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Inferring link metrics from end-to-end path measurements: identifiability and monitor placement," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 4, pp. 1351–1368, 2014.

[6] W. Dong, Y. Gao, W. Wu, J. Bu, C. Chen, and X. Y. Li, "Optimal monitor assignment for preferential link tomography in communication networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 210–223, 2017.

[7] R. Yang, C. Feng, L. Wang, W. Wu, K. Wu, J. Wang, and Y. Xu, "On the optimal monitor placement for inferring additive metrics of interested paths," in *IEEE INFOCOM*, Honolulu, HI, April 2018.

[8] Y. Lin, T. He, S. Wang, K. Chan, and S. Pasteris, "Looking glass of nfv: inferring the structure and state of nfv network from external observations," *IEEE/ACM Transactions on Networking*, 2020.

[9] C. Feng, J. An, K. Wu, and J. Wang, "Bound inference and reinforcement learning-based path construction in bandwidth tomography," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2021.

[10] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[11] Y. Qiao, K. Wu, and M. Khabbazian, "Non-intrusive and high-efficient balance tomography in the lightning network," in *ACM ASIACCS*, Hong Kong, June 2021.

[12] L. Ma, T. He, K. K. Leung, A.Swami and D.Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *Proceedings of the 2013 Conference on Internet Measurement Conference*. New York, NY, USA: ACM, 2013, pp. 391–404.

[13] J. Xie and X. Wang, "A survey of mobility management in hybrid wireless mesh networks," *IEEE network*, vol. 22, no. 6, pp. 34–40, 2008.

[14] Y. Liu, K.-F. Tong, X. Qiu, Y. Liu, and X. Ding, "Wireless mesh networks in iot networks," in *2017 International workshop on electromagnetics: applications and student innovation competition*. IEEE, 2017, pp. 183–185.

[15] J. Burchard, D. Chemodanov, J. Gillis, and P. Calyam, "Wireless mesh networking protocol for sustained throughput in edge computing," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 958–962.

[16] D. Zhang, M. Piao, T. Zhang, C. Chen, and H. Zhu, "New algorithm of multi-strategy channel allocation for edge computing," *AEU-International Journal of Electronics and Communications*, vol. 126, p. 153372, 2020.

[17] A. A. Pirzada and M. Portmann, "High performance aodv routing protocol for hybrid wireless mesh networks," in *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*. IEEE, 2007, pp. 1–5.

[18] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[19] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in ip networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1092–1103, 2006.

[20] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2196–2209, 2006.

[21] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 906–916, 2012.

[22] Gopalan, Abishek and S.Ramasubramanian, "On the maximum number of linearly independent cycles and paths in a network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1373–1388, 2014.

[23] L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*, 2013, pp. 581–590.

[24] Y. Gao, W. Dong, W. Wu, C. Chen, X. Y. Li, and J. Bu, "Scalpel: scalable preferential link tomography based on graph trimming," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1392–1403, 2016.

[25] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Monitor placement for maximal identifiability in network tomography," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 1447–1455.

[26] S. S. Chaudhari and R. C. Biradar, "Survey of bandwidth estimation techniques in communication networks," *wireless personal communications*, vol. 83, no. 2, pp. 1425–1476, 2015.

[27] M. Jain and C. Dovrolis, "Pathload: a measurement tool for end-to-end available bandwidth," in *In Proceedings of Passive and Active Measurements (PAM) Workshop*. Citeseer, 2002.

[28] T. Yang, Y. Jin, Y. Chen, and Y. Jin, "Rt-wabest: A novel end-to-end bandwidth estimation tool in ieee 802.11 wireless network," *International Journal of Distributed Sensor Networks*, vol. 13, no. 2, p. 1550147717694889, 2017.

[29] F. Boccardi, J. Andrews, H. Elshaer, M. Dohler, S. Parkvall, P. Popovski, and S. Singh, "Why to decouple the uplink and downlink in cellular networks and how to do it," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 110–117, 2016.

[30] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," in *ATM Forum contribution*, vol. 99, no. 45, 1999.

[31] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, "Optimal tradeoff between sum-rate efficiency and jain's fairness index in resource allocation," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3496–3509, 2013.

[32] F. Zabini, A. Bazzi, B. M. Masini, and R. Verdone, "Optimal performance versus fairness tradeoff for resource allocation in wireless systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2587–2600, 2017.

[33] T. He, L. Ma, A. Swami, and D. Towsley, *Network tomography: identifiability, measurement design, and network state inference*. Cambridge University Press, 2021.

[34] E. H. Ong and J. Y. Khan, "On optimal network selection in a dynamic multi-rat environment," *IEEE Communications Letters*, vol. 14, no. 3, pp. 217–219, 2010.

APPENDIX

Algorithm 3: ϵ-LUAN(v): Locally Updating links At Node v with tolerance ϵ.

input : ϵ, $\mathcal{L}_v$, $\mathcal{L}_v^I$. For each $l_k \in \mathcal{L}_v \setminus \mathcal{L}_v^I$, $[\hat{l}_{k-}, \hat{l}_{k+}]$

output: For each $l_i \in \mathcal{L}_v \setminus \mathcal{L}_v^I$, updating its bandwidth bound based on the results of CBLN(l_i, v). In addition, according to corresponding $\mathcal{C}_{l_{i-}}(v)$ and $\mathcal{C}_{l_{i+}}(v)$, updating $l^-_{v(j,i)}$ and $l^+_{v(j,i)}$ at each $l_j \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}_v^I$.

begin
  Adding all links in $\mathcal{L}_v \setminus \mathcal{L}_v^I$ to the list L;
  while L *is not empty* **do**
    $l_i \leftarrow L.pop(0)$;
    Running CBLN(l_i, v) and obtaining $[\hat{l}'_{i-}, \hat{l}'_{i+}]$, $\mathcal{C}_{i-}(v)$, $\mathcal{C}_{i+}(v)$;
    if $|\hat{l}'_{i-} - \hat{l}_{i-}| + |\hat{l}'_{i+} - \hat{l}_{i+}| \geq \epsilon$ **then**
      Updating current bound of l_i to $[\hat{l}'_{i-}, \hat{l}'_{i+}]$;
      for $l_j \in (\mathcal{L}_v \setminus l_i) \setminus \mathcal{L}_v^I$ **do**
        Updating $l^-_{v(j,i)}$, $l^+_{v(j,i)}$ based on $\mathcal{C}_{i-}(v)$, $\mathcal{C}_{i+}(v)$;
      end
      if l_i *becomes identifiable* **then**
        Adding l_i to $\mathcal{L}_v^I$;
      end
    end
    for $l_k \in L$ **do**
      if $\hat{l}_{k-} == \hat{l}_{i-}$ *and* $\hat{l}_{k+} == \hat{l}_{i+}$ *and* $|\hat{l}'_{i-} - \hat{l}_{k-}| + |\hat{l}'_{i+} - \hat{l}_{k+}| \geq \epsilon$ **then**
        Removing l_k from L;
        Updating current bound of l_k to $[\hat{l}'_{i-}, \hat{l}'_{i+}]$;
        $l^-_{v(i,k)} = l^-_{v(k,i)}$;$l^+_{v(i,k)} = l^+_{v(k,i)}$;
        for $l_j \in (\mathcal{L}_v \setminus (l_k \cup l_i)) \setminus \mathcal{L}_v^I$ **do**
          $l^-_{v(j,k)} = l^-_{v(j,i)}$; $l^+_{v(j,k)} = l^+_{v(j,i)}$;
        end
        if l_k *becomes identifiable* **then**
          Adding l_k to $\mathcal{L}_v^I$;
        end
      end
    end
  end
end